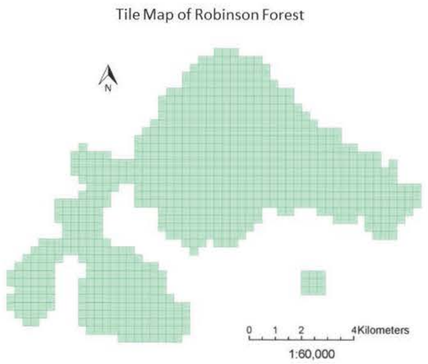
# Distributed Object Segmentation in Big Spatial Data

Hamid Hamraz, Ph.D Candidate
Department of Computer Science, University of Kentucky
hhamraz@cs.uky.edu www.cs.uky.edu/~hhamraz/

## Problem Overview

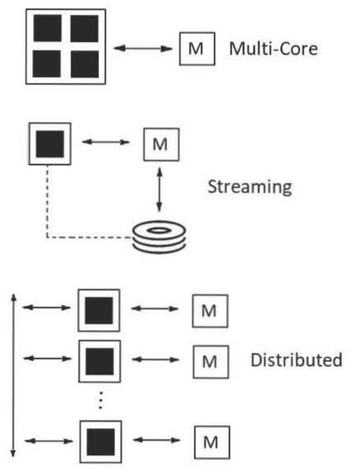
- Big spatial data is delivered in the shape of several tiles and requires distributed computing.
- Object segmentation is not perfectly parallel because the objects may lie across the tiles.




1

## Related Work

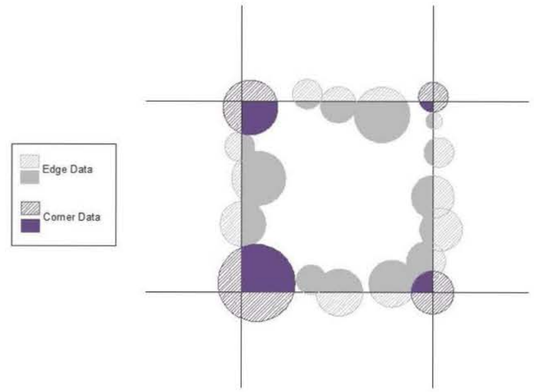
- Multi-core processing
  - Limited memory and cores
  - Limits the size of problem
- Streaming
  - Manages memory
  - Does not help with the time
- Overlapping data units for distributed computing
  - Increases the problem size
  - How to merge overlapping result?




2

## Proposed Approach

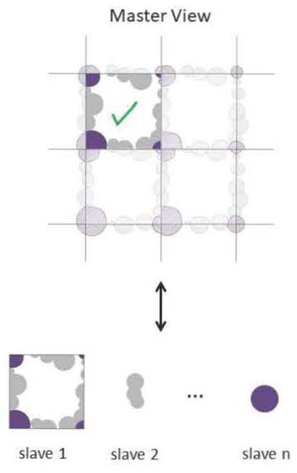
- Upon segmenting a data unit, identify boundary data
  - Edge
  - Corner




3

## Master-Slave Processing

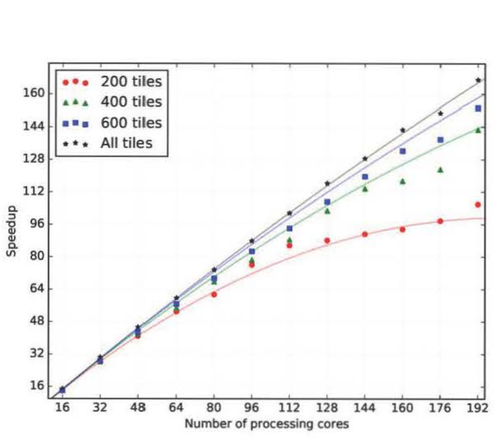
- Master maintains a global picture
  - Assign the slaves to segment units
  - Joins the boundary data once ready and send them to slaves for re-segmentation
- Slaves segment units
  - Send the boundary data to the master
  - Re-segment joined boundary data as directed by the master




4

## Segmenting Trees of Entire Robinson Forest

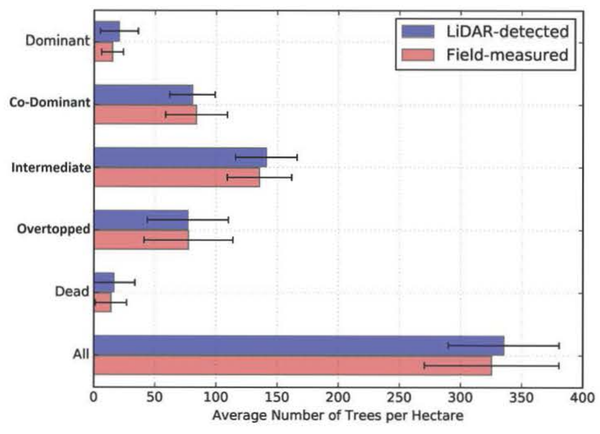
- Implemented using MPI
- Run on the UK HPC cluster
- Segmented nearly two million trees
  - 3 hours using 160 processors
  - 144 times faster than a single processor




Experimental (Symbols) vs. Theoretical (continuous lines)

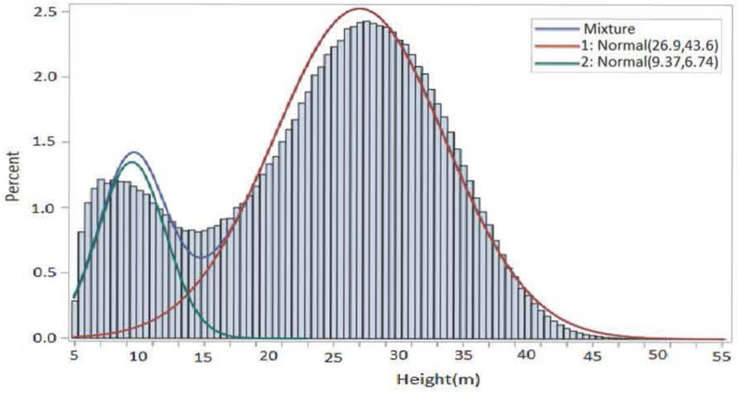
5

## Estimating tree numbers based on two million segmented trees




6

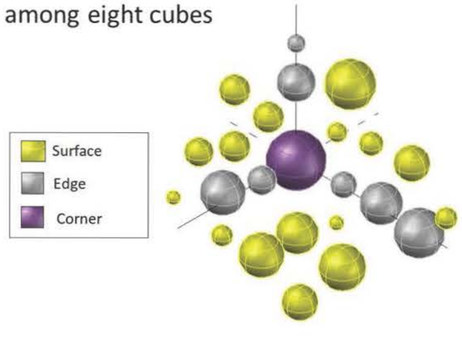
## Distribution of Tree Height




7

## Generalization to 3D

- Upon segmenting a cube, boundary data are identified
  - Surface: shared among two cubes
  - Edge: shared among four cubes
  - Corner: shared among eight cubes




8

## Implementation using MapReduce

- Map phase
  - Segment a unit and identify the boundary data
  - Assign a unique key to each of the boundary data, which must be the same across all units sharing the boundary
- Reduce Phase
  - Join all the boundary data pieces provided and re-segment
- No master needed
- Easier to implement once you know MapReduce
- Not as efficient as MPI

9